

# LE PROBLÈME DES 1000 PORTES

Le problème est le suivant :

Vous êtes dans un couloir contenant 1 000 portes (numérotées de 0 à 999). Chaque porte peut être soit ouverte soit fermée (et "l'état" d'une porte est donc son caractère ouvert ou fermé). Au départ, toutes les portes sont fermées. On effectue alors la suite d'opérations suivantes :

- On change d'état toutes les portes dont le numéro est un multiple de 1
- On change d'état toutes les portes dont le numéro est un multiple de 2
- ...
- On change d'état toutes les portes dont le numéro est un multiple de 1 000

Vous allez alors être téléporté devant une porte de ce couloir, celle de votre choix. Si cette porte est la 23<sup>e</sup> porte ouverte (en comptant par ordre croissant), votre vœu le plus cher s'exauce (il se met à neiger et tous les cours sont annulés!) sinon votre pire cauchemar prend forme (vous devenez professeur(e) d'informatique...)

Nous allons essayer ensemble de faire pour le mieux. On va pour cela se servir d'un tableau, ainsi que de boucles.

Partie A : sur papier.

1. Dans notre algorithme, on va utiliser une variable "portes" de type tableau de booléens. On adoptera alors la convention suivante dans la suite de l'algorithme :

portes[i] vaut Vrai quand la porte numéro i est ouverte (et Faux sinon)

Déclarer la variable globale "portes", et l'initialiser selon l'état des 1 000 portes au début du problème.

2. Ecrire un algorithme « étape » qui a la spécification suivante :
  - il prend en entrée un nombre entier  $k$  entre 1 et 1 000
  - il utilise la variable globale "portes" contenant les états des 1 000 portes du couloir, et il change ce tableau de manière à changer d'état toutes les portes dont le numéro est un multiple de  $k$
3. A l'aide d'une boucle et de l'algorithme de la question précédente, écrire un algorithme « état\_final » qui utilise la variable globale "portes", et qui, une fois terminé, a mis à jour ce tableau pour donner l'état des 1 000 portes juste avant la téléportation.
4. Compléter l'algorithme de manière à faire trouver la porte menant à la neige salvatrice.

Partie B : sous Python.

1. Déclarer une variable "portes" de type tableau contenant 1 000 cases, et initialiser toutes les cases à `False` : cela se fait avec la commande :

`portes = 1000*[False]`

2. Ecrire l'algorithme précédent dans Python. On notera que pour savoir si un nombre est multiple de  $k$ , il suffit de savoir si le reste de sa division euclidienne par  $k$  vaut 0.
3. Faire tourner l'algorithme pour trouver la porte tant souhaitée.

Complément : boucles sous Python.

Une boucle de type `for` en Python s'écrit "for i in [liste de choses]". On crée une liste de nombres avec l'instruction `range([debut], fin, [pas])`. Si début n'est pas fixé, il vaut 0 et si pas n'est pas fixé, il vaut 1. Ex :

`range(10) = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`

`range(4, 10) = [4, 5, 6, 7, 8, 9]`

`range(4, 10, 2) = [4, 6, 8]`